

/*

* Copyright 2018 Google Inc. All Rights Reserved.

*

* Licensed under the Apache License, Version 2.0 (the "License");

* you may not use this file except in compliance with the License.

* You may obtain a copy of the License at

*

* <http://www.apache.org/licenses/LICENSE-2.0>

*

* Unless required by applicable law or agreed to in writing, software

* distributed under the License is distributed on an "AS IS" BASIS,

* WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

* See the License for the specific language governing permissions and

* limitations under the License.

*/

package com.google.ar.core.examples.java.computervision;

import android.graphics.Bitmap;

import android.graphics.ImageFormat;

import android.graphics.YuvImage;

import android.graphics.drawable.BitmapDrawable;

import android.graphics.drawable.Drawable;

import android.media.Image;

import android.opengl.GLES20;

import android.opengl.GLSurfaceView;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.util.Log;

import android.util.Size;

import android.view.Gravity;

```
import android.view.View;

import android.widget.CompoundButton;

import android.widget.RadioButton;

import android.widget.RadioGroup;

import android.widget.Switch;

import android.widget.TextView;

import android.widget.Toast;

import com.google.ar.core.ArCoreApk;

import com.google.ar.core.Camera;

import com.google.ar.core.CameraConfig;

import com.google.ar.core.CameraConfigFilter;

import com.google.ar.core.CameraIntrinsics;

import com.google.ar.core.Config;

import com.google.ar.core.Frame;

import com.google.ar.core.Session;

import com.google.ar.core.examples.java.common.helpers.CameraPermissionHelper;

import com.google.ar.core.examples.java.common.helpers.FullScreenHelper;

import com.google.ar.core.examples.java.common.helpers.SnackbarHelper;

import com.google.ar.core.examples.java.common.helpers.TrackingStateHelper;

import com.google.ar.core.exceptions.CameraNotAvailableException;

import com.google.ar.core.exceptions.NotYetAvailableException;

import com.google.ar.core.exceptions.UnavailableApkTooOldException;

import com.google.ar.core.exceptions.UnavailableArcoreNotInstalledException;

import com.google.ar.core.exceptions.UnavailableSdkTooOldException;

import com.google.ar.core.exceptions.UnavailableUserDeclinedInstallationException;


import java.io.ByteArrayInputStream;

import java.io.ByteArrayOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.nio.ByteBuffer;
```

```

import java.util.ArrayList;

import java.util.Collections;

import java.util.EnumSet;

import java.util.List;

import javax.microedition.khronos.egl.EGLConfig;

import javax.microedition.khronos.opengles.GL10;


/** This is a simple example that demonstrates CPU image access with ARCore. */

public class ComputerVisionActivity extends AppCompatActivity implements
GLSurfaceView.Renderer {

    private static final String TAG = ComputerVisionActivity.class.getSimpleName();

    private static final String CAMERA_INTRINSICS_TEXT_FORMAT =

        "\tUnrotated Camera %s %s Intrinsics:\n\tFocal Length: (%.2f, %.2f)"
        + "\n\tPrincipal Point: (%.2f, %.2f)"
        + "\n\t%s Image Dimensions: (%d, %d)"
        + "\n\tUnrotated Field of View: (%.2f°, %.2f°)"
        + "\n\tRender frame time: %.1f ms (%.0ffps)"
        + "\n\tCPU image frame time: %.1f ms (%.0ffps)";

    private static final float RADIANS_TO_DEGREES = (float) (180 / Math.PI);


    // This app demonstrates two approaches to obtaining image data accessible on CPU:
    // 1. Access the CPU image directly from ARCore. This approach delivers a frame without latency
    //   (if available), but currently is lower resolution than the GPU image.
    // 2. Download the texture from GPU. This approach incurs a 1-frame latency, but allows a high
    //   resolution image.

    private enum ImageAcquisitionPath {

        CPU_DIRECT_ACCESS,

        GPU_DOWNLOAD

    }

```